Minería de datos y Patrones

Version 2024-1

Object Detection with SIFT

[Capítulo 2]

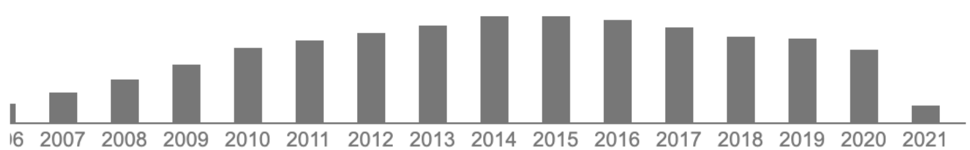
Dr. José Ramón Iglesias
DSP-ASIC BUILDER GROUP
Director Semillero TRIAC
Ingenieria Electronica
Universidad Popular del Cesar

SIFT : Scale Invariant Feature Transform

David Lowe

Distinctive image features from scale-invariant keypoints

Authors	David G Lowe
Publication date	2004/11/1
Journal	International journal of computer vision
Volume	60
Issue	2
Pages	91-110
Publisher	Springer Netherlands
Description	This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many images. This paper also describes an approach to using these features for object recognition. The recognition proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm, followed by a Hough transform to identify clusters belonging to a single object, and finally performing verification through …
Total citations	Cited by 62027



D. G. Lowe. [Distinctive image features from scale-invariant features](). Intl. Journal of Computer Vision, 60(2):91-110, 2004.

Why SIFT?

SIFT : Scale Invariant Feature Transform

keypoints

description

128

17	26	41	65	104	167	5	23	11
46	71	112	178	285	282	24	46	26
125	194	307	489	608	635	57	107	69
340	531	842	1168	1531	1232	166	242	174
5	8	13	21	34	55	1	7	3
6	9	14	22	35	56	2	8	4
17	26	41	65	104	167	5	23	11
46	71	112	178	285	282	24	46	26
125	194	307	489	608	635	57	107	69
340	531	842	1168	1531	1232	166	242	174

N

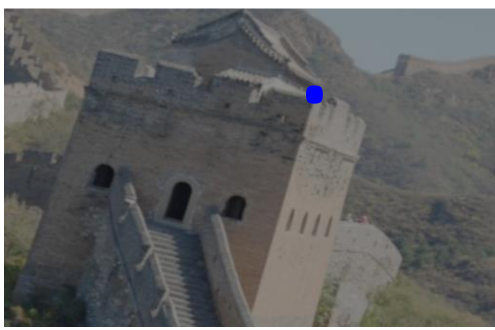
Each keypoint is described as a 128-element vector

keypoints → description →

17	26	41	65	104	167	5	23	11
46	71	112	178	285	282	24	46	26
125	194	307	489	608	635	57	107	69
340	531	842	1168	1531	1232	166	242	174
5	8	13	21	34	55	1	7	3
6	9	14	22	35	56	2	8	4
17	26	41	65	104	167	5	23	11
46	71	112	178	285	282	24	46	26
125	194	307	489	608	635	57	107	69
340	531	842	1168	1531	1232	166	242	174

keypoints → description →

3	4	6	9	14	22	35	56	2
8	11	17	26	41	65	104	167	5
21	30	46	71	112	178	285	282	24
56	81	125	194	307	489	608	635	57
149	220	340	531	842	1168	1531	1232	166
2	3	5	8	13	21	34	55	1
3	4	6	9	14	22	35	56	2
8	11	17	26	41	65	104	167	5
21	30	46	71	112	178	285	282	24
56	81	125	194	307	489	608	635	57

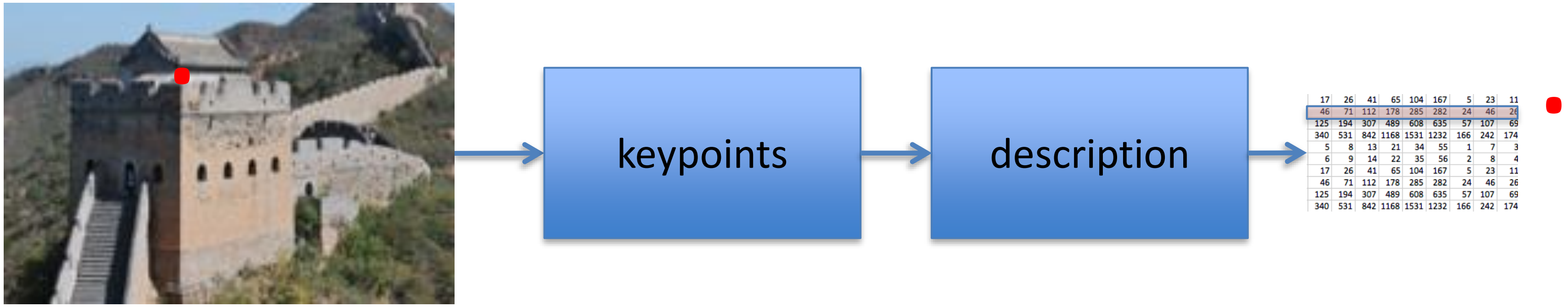


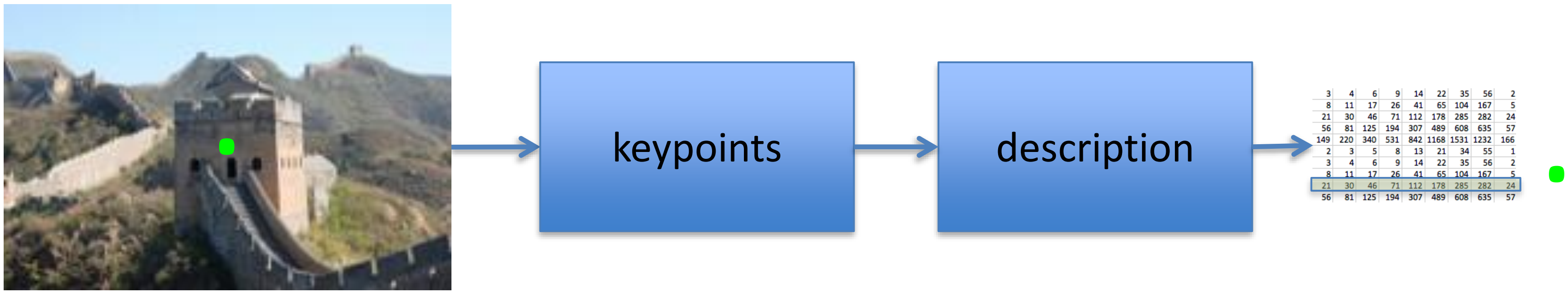
keypoints → description →

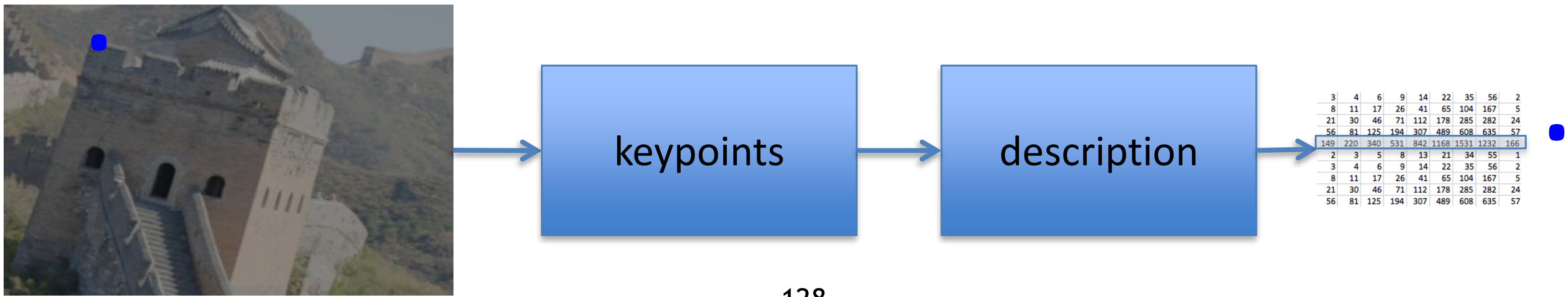
3	4	6	9	14	22	35	56	2
8	11	17	26	41	65	104	167	5
21	30	46	71	112	178	285	282	24
56	81	125	194	307	489	608	635	57
149	220	340	531	842	1168	1531	1232	166
2	3	5	8	13	21	34	55	1
3	4	6	9	14	22	35	56	2
8	11	17	26	41	65	104	167	5
21	30	46	71	112	178	285	282	24
56	81	125	194	307	489	608	635	57

128

1	3	5	17	23	21	45	11	15	7	67	56	12	87	56	34	12	15	19	9	7	79	19	...	25







128

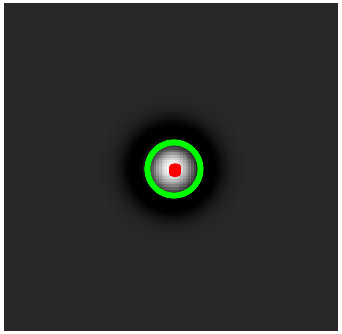
1	3	5	17	23	21	45	11	15	7	67	56	12	87	56	34	12	15	19	9	7	79	19	...	25
23	65	32	90	76	56	34	98	6	8	56	8	9	23	8	45	2	43	67	19	62	78	55	...	7
99	76	34	12	98	76	90	55	43	87	65	43	32	65	7	9	1	3	45	66	39	18	39	...	78

SIFT

- It is used to detect keypoints
- Each keypoint is described using a 128-element vector called ‘SIFT-descriptor’
- SIFT-descriptor is:
    - Scale invariant
    - Rotation invariant
    - Illumination invariant
    - Viewpoint invariant
- SIFT-descriptor is like a ‘signature’:
    - SIFT-descriptors of the same point (in different images) are very similar.
    - SIFT-descriptors of different points are very different.

Detection of Keypoints

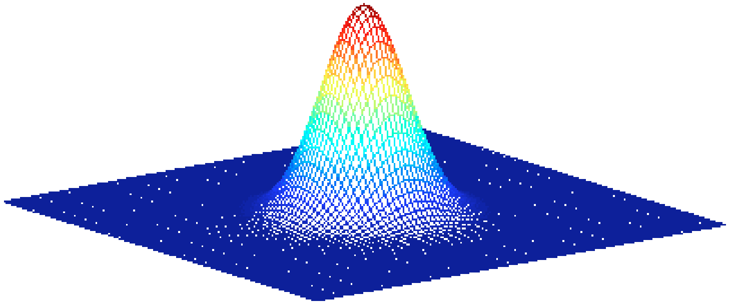
A synthetic image with a spot



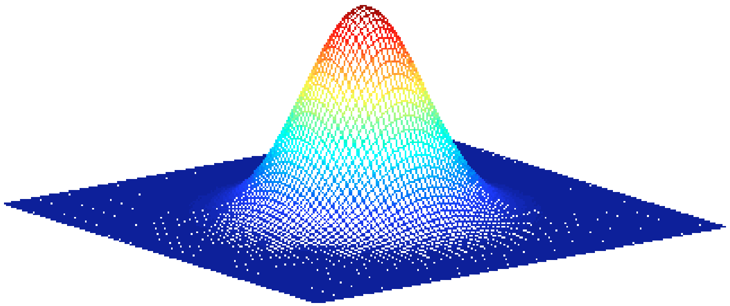
Two goals:

- Where?
- Size?

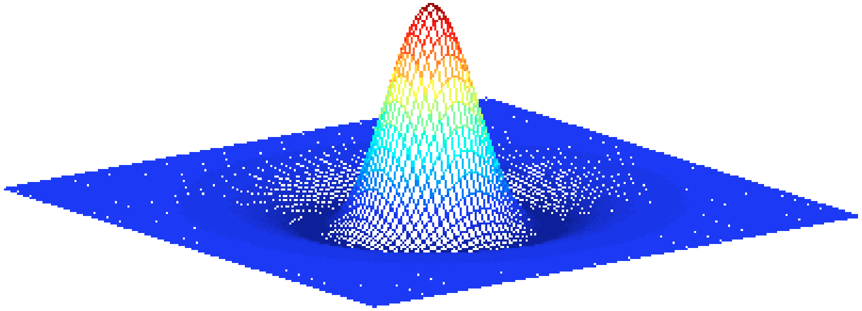
For the detection a DoG – Mask is used:



G1 = Gaussian 1 (with σ)

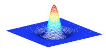
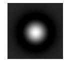
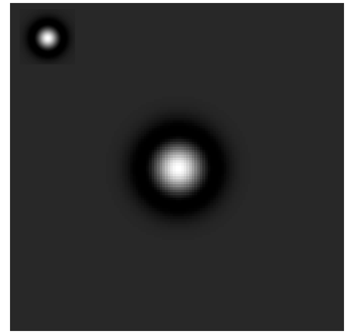


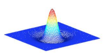
G2 = Gaussian 2 (with *k*σ)

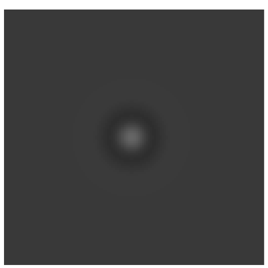


Difference of Gaussians (DoG): G2-G1

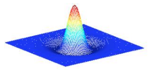
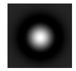
DoG Mask

  ★ 

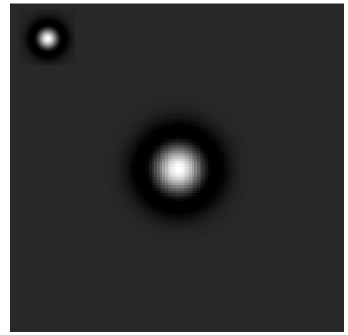


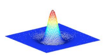
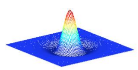


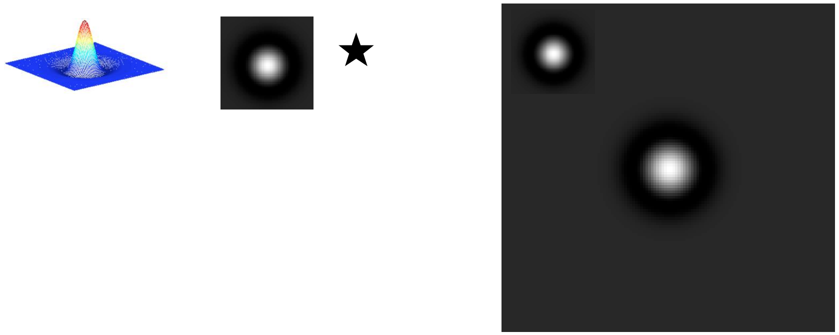
★ : Convolution

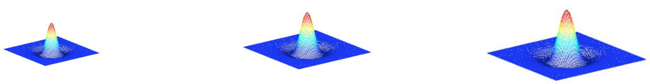



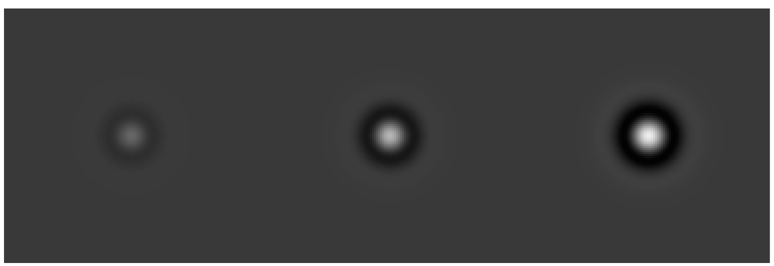
$\star$



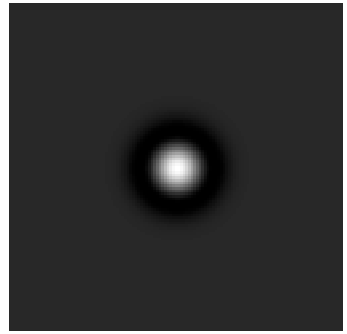





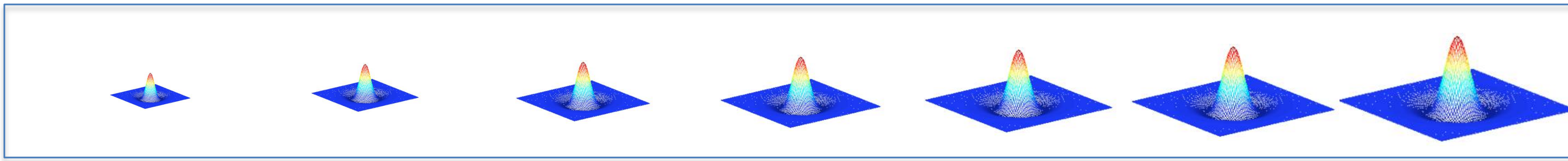




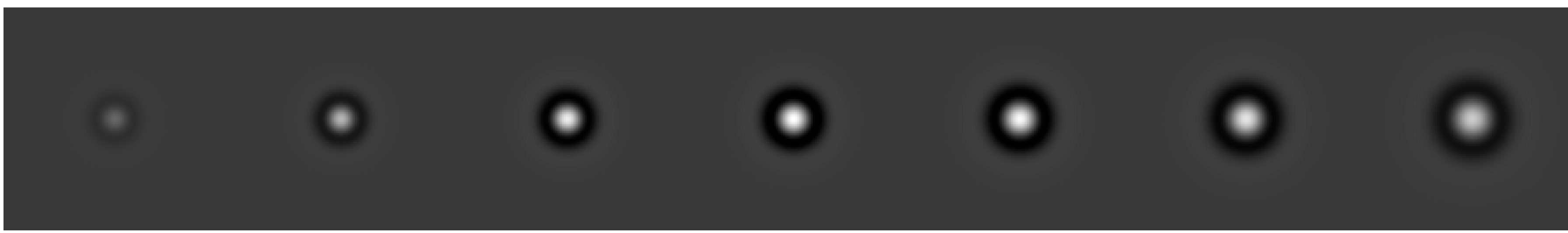
Input



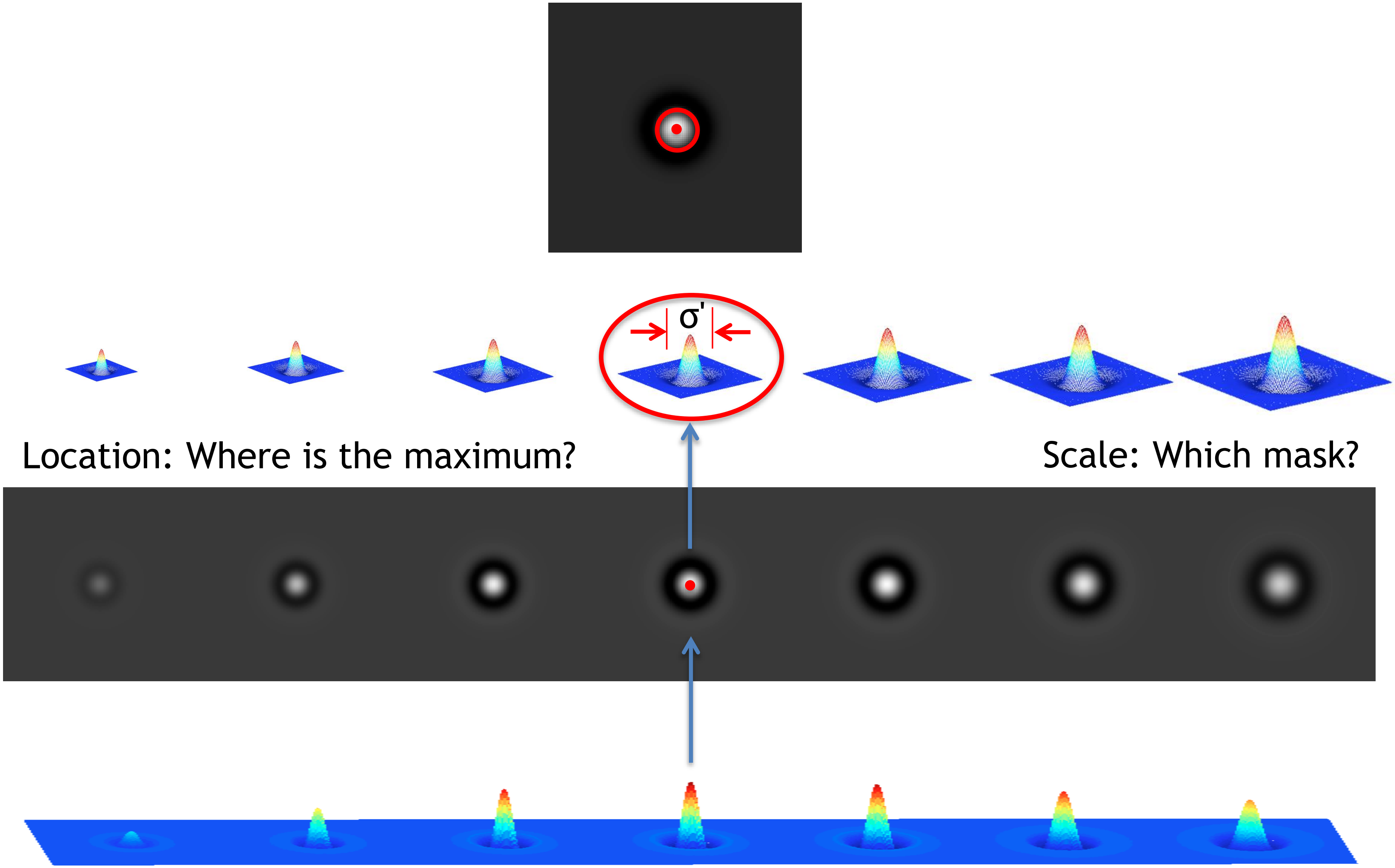
★ with DoG of different σ



=



Outputs


σ'
Location: Where is the maximum?
Scale: Which mask?

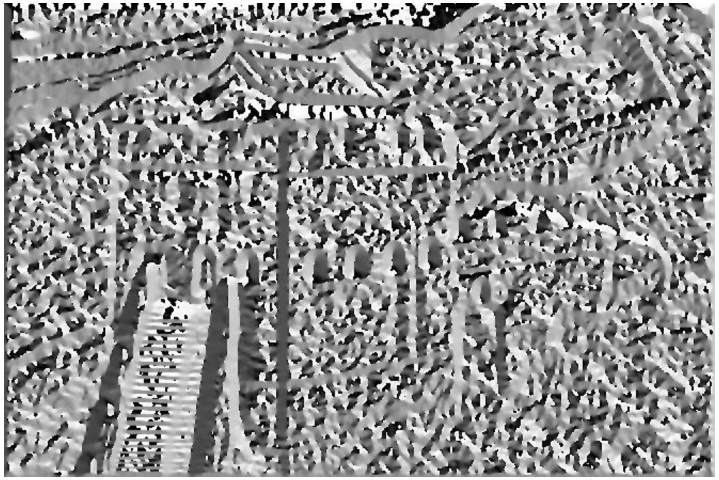
Descriptor of a Keypoint

First, we have to understand
what is a Histogram of Gradients

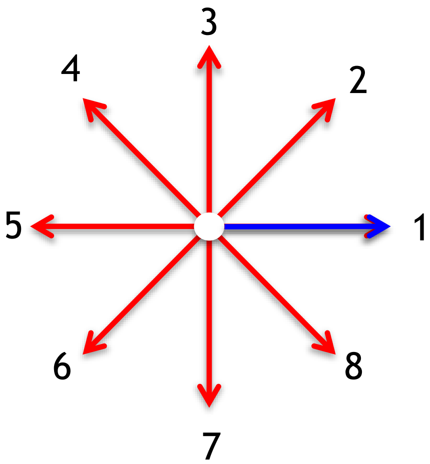
Histogram of Gradients

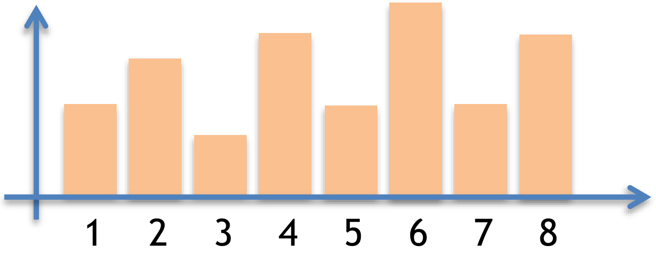
R: Magnitude

A: Angle



Histogram of 8 directions

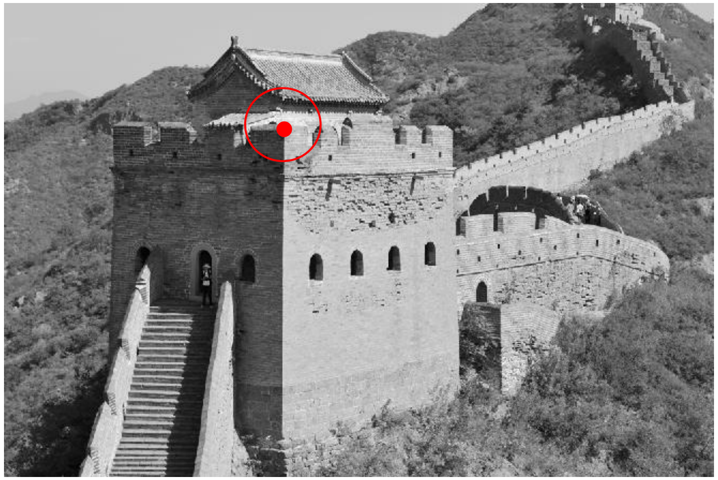




Now, we can understand how to build the SIFT descriptor

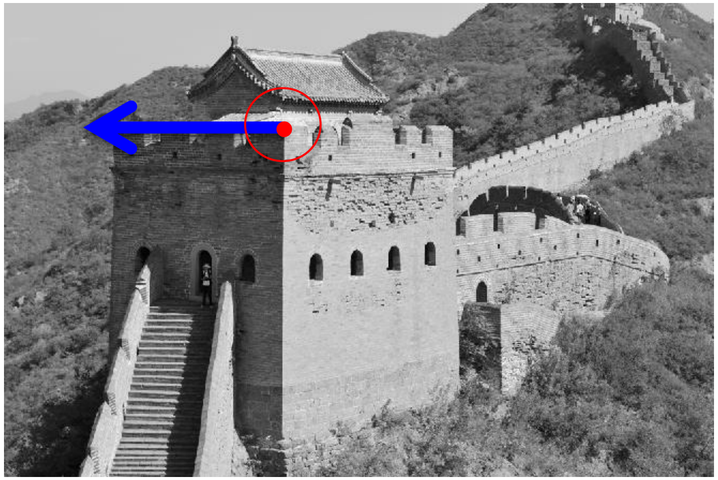
How to build the descriptor?

1. Find a keypoint (x,y,σ’)



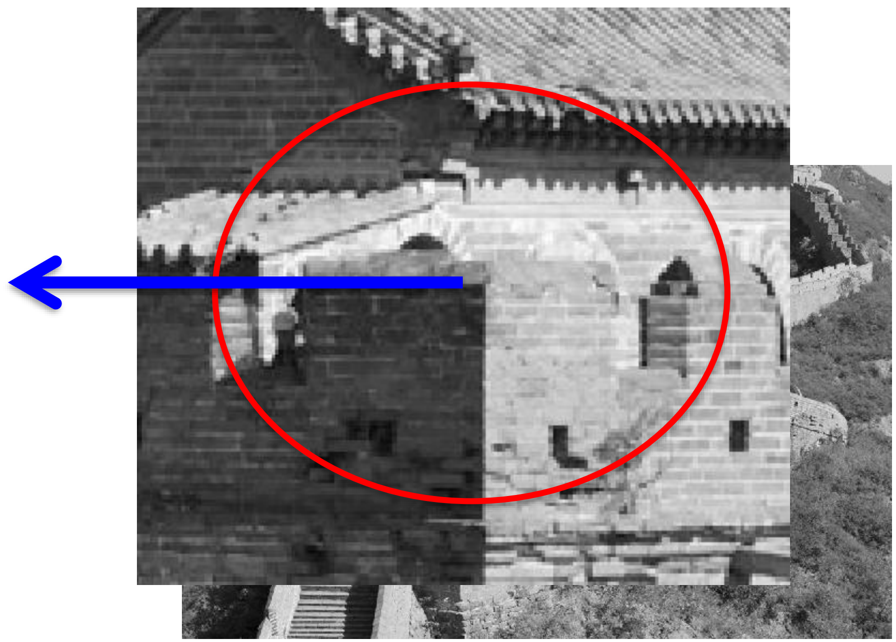
How to build the descriptor?

1. Find a keypoint (x,y,σ')
2. Find the orientation using A(x,y) matrix



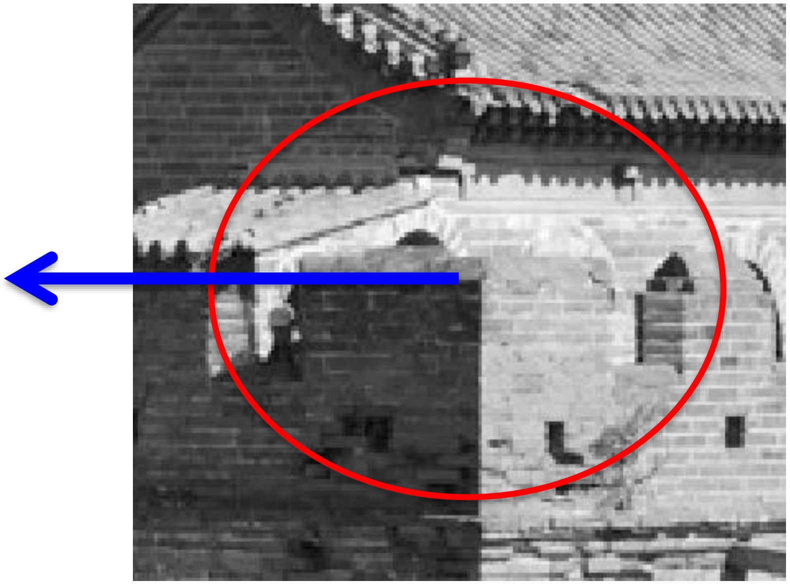
How to build the descriptor?

1. Find a keypoint (x,y,σ’)
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ’.

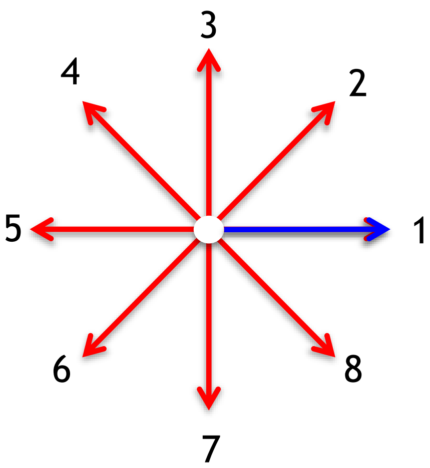


How to build the descriptor?

1. Find a keypoint (x,y,σ')
2. Find the orientation using $A(x,y)$ matrix
3. Take a window centered in the keypoint of size $1.5\ \sigma'$.
4. Align the window to direction '1'.

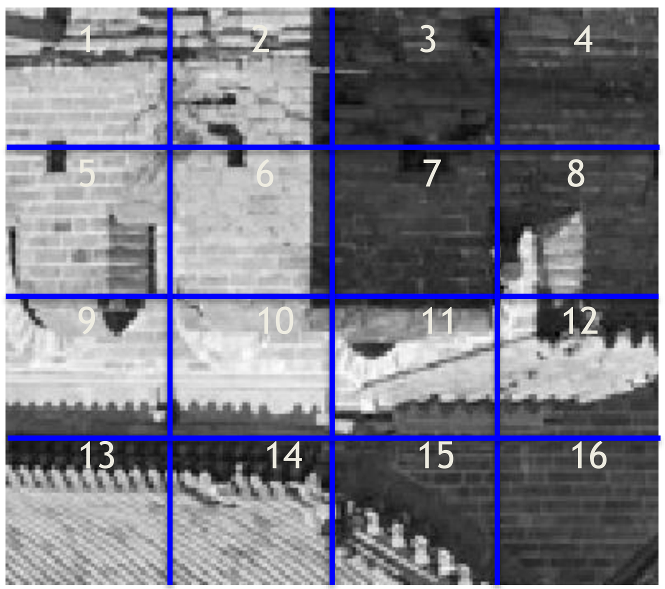


8 directions

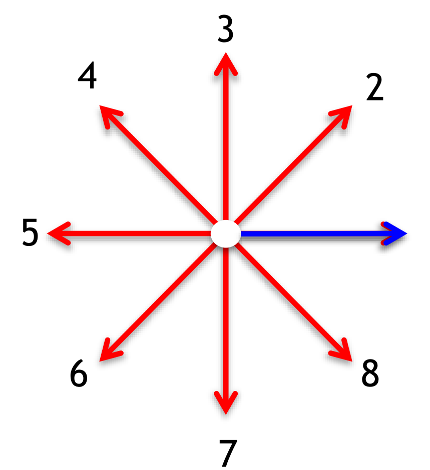


How to build the descriptor?

1. Find a keypoint (x,y,σ')
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ'.
4. Align the window to direction '1'.



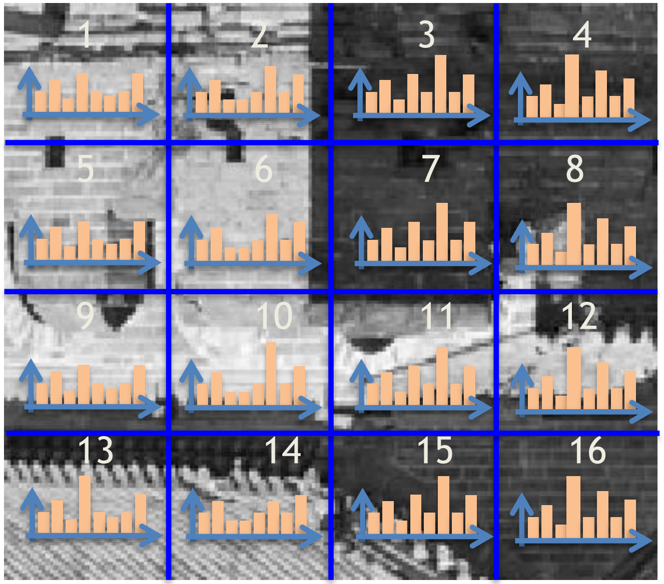
8 directions

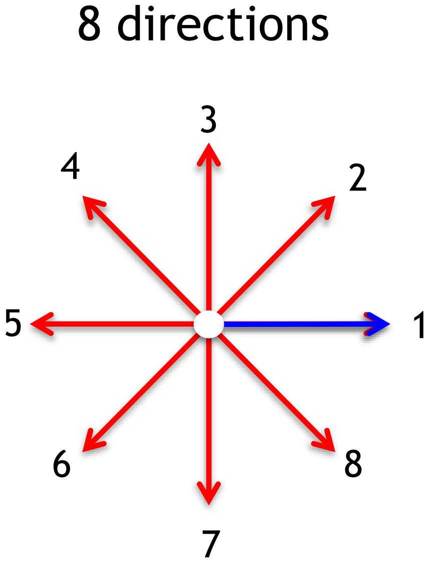


5. Define 16 partitions

How to build the descriptor?

1. Find a keypoint (x,y,σ’)
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ’.
4. Align the window to direction ‘1’.





5. Define 16 partitions
6. Compute 8 bin histograms in each partition

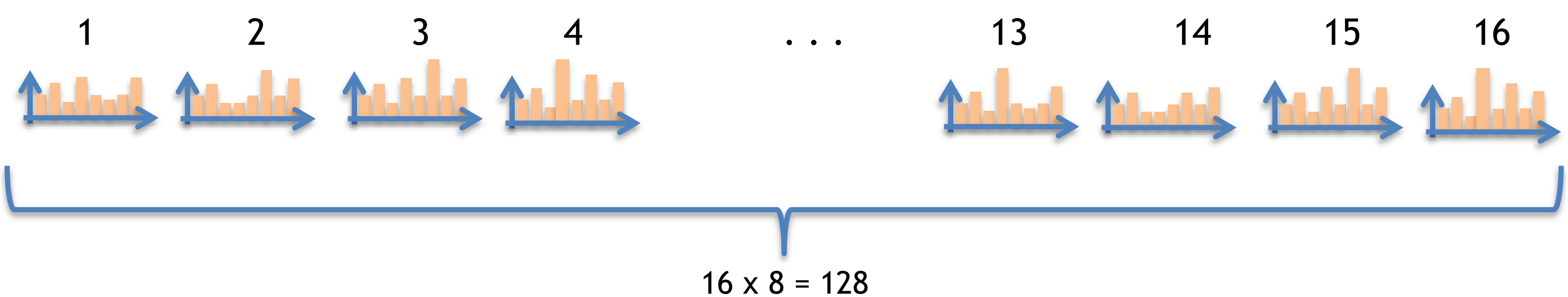
How to build the descriptor?

1. Find a keypoint (x,y,σ’)
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ’.
4. Align the window to direction ‘1’.

5. Define 16 partitions
6. Compute 8 bin histograms in each partition

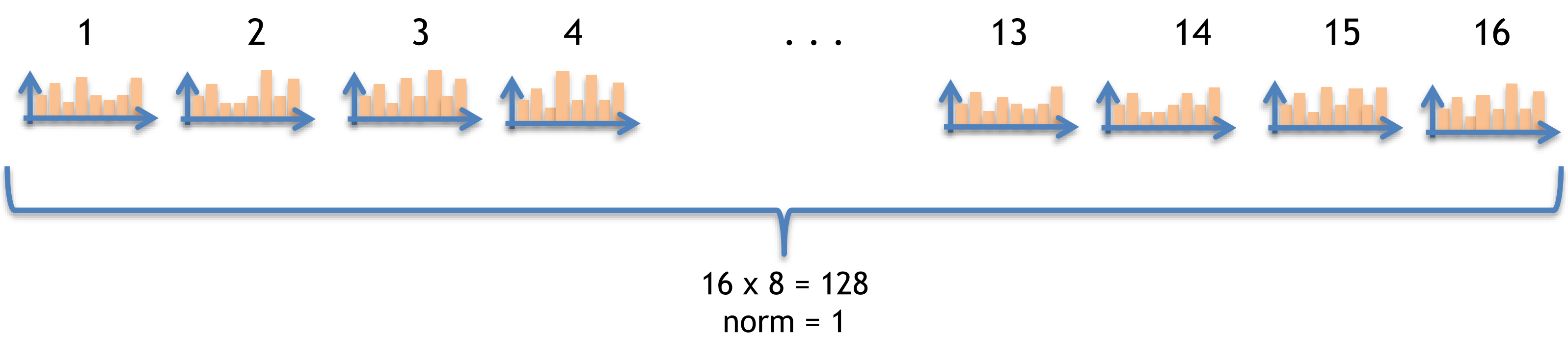
How to build the descriptor?

7. Concatenate all histograms



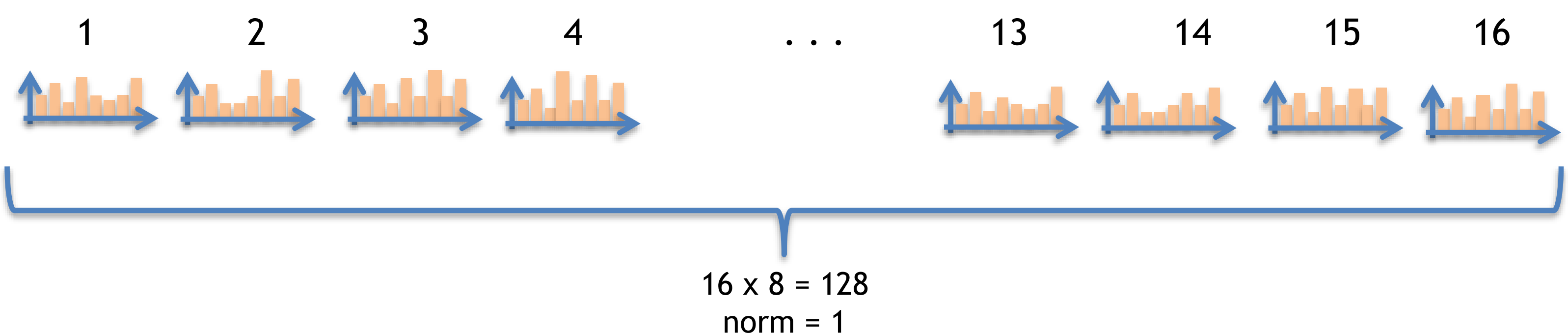
How to build the descriptor?

7. Concatenate all histograms
8. Normalize

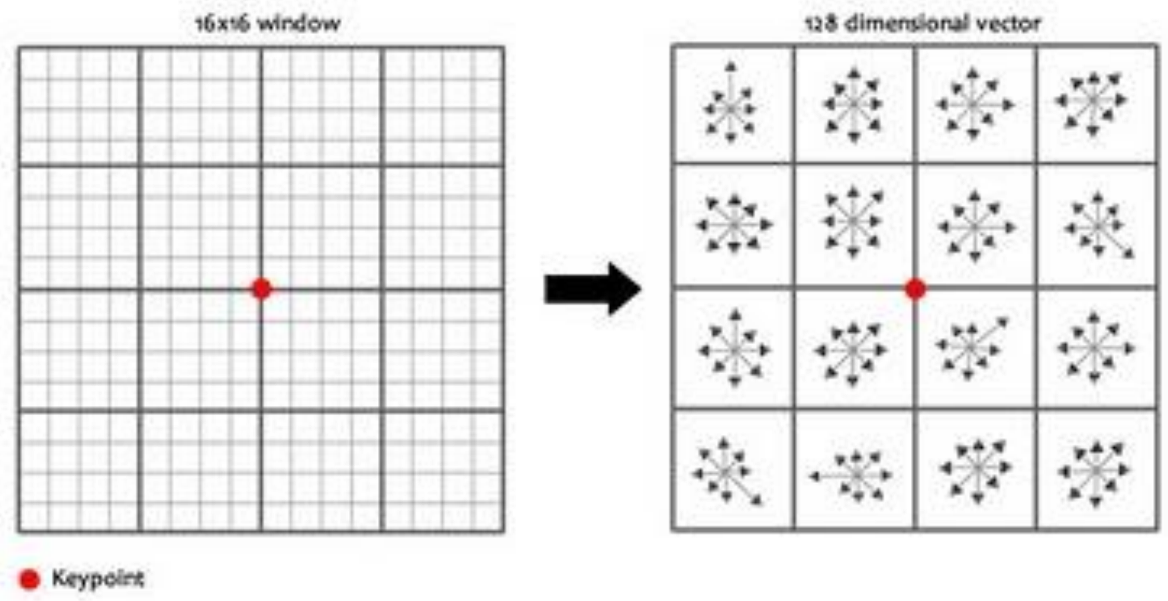


How to build the descriptor?

7. Concatenate all histograms
8. Normalize
9. The End



SIFT : Scale Invariant Feature Transform



© Lowe

Each keypoint is described as a 128-element vector

A. Saharkhiz: Details behind SIFT
http://areshmatlab.blogspot.com/2010/07/details-behind-sift-feature-detection.html

- Why is SIFT-descriptor … ?
  - Scale invariant
  - Rotation invariant
  - Illumination invariant
  - Viewpoint invariant

Why is SIFT-descriptor … ?

a) Scale invariant
b) Rotation invariant
c) Illumination invariant
d) Viewpoint invariant

1. Find a keypoint (x,y,σ')
2. Find the orientation using A(x,y) matrix
3. Take a window centered in the keypoint of size 1.5 σ'.
4. Align the window to direction '1'.
5. Define 16 partitions
6. Compute 8 bin histograms in each partition
7. Concatenate all histograms
8. Normalize

Example

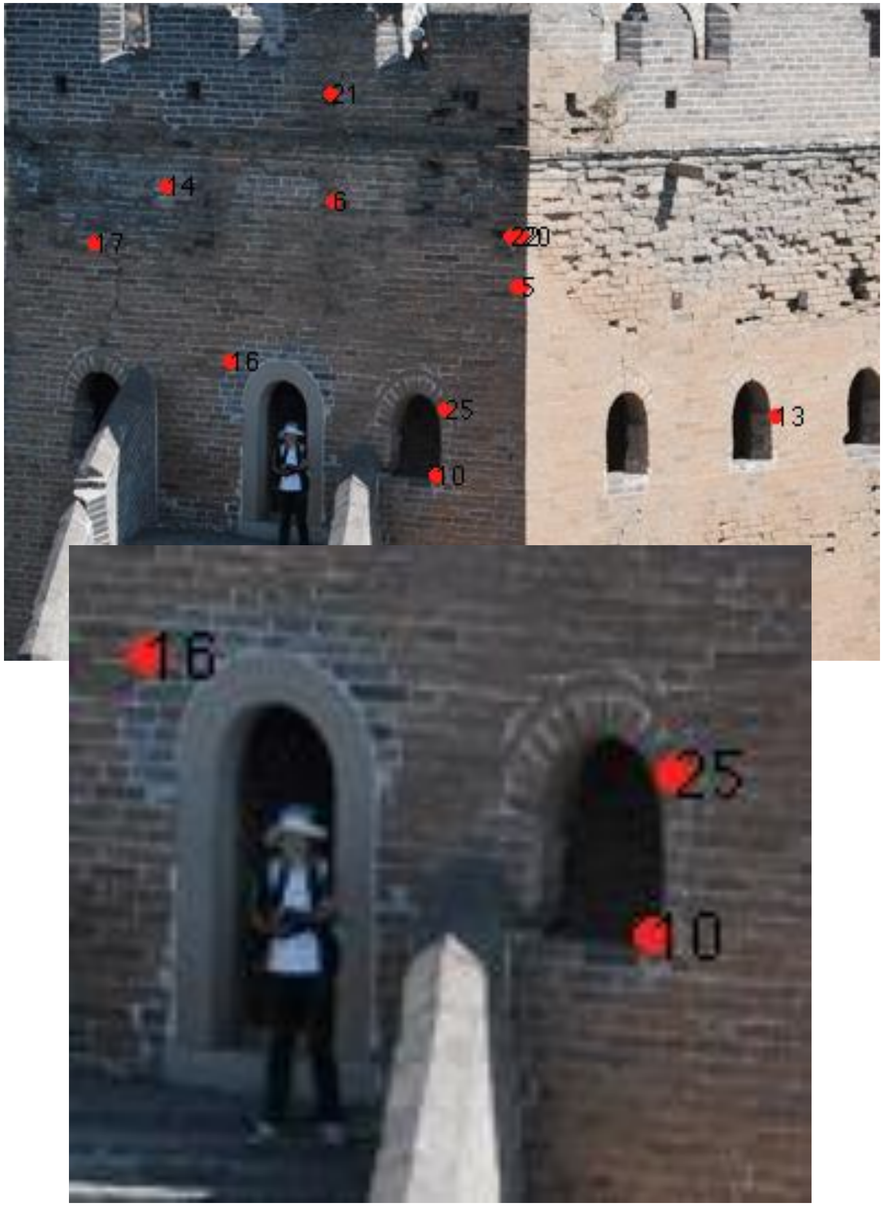
SIFT keypoints

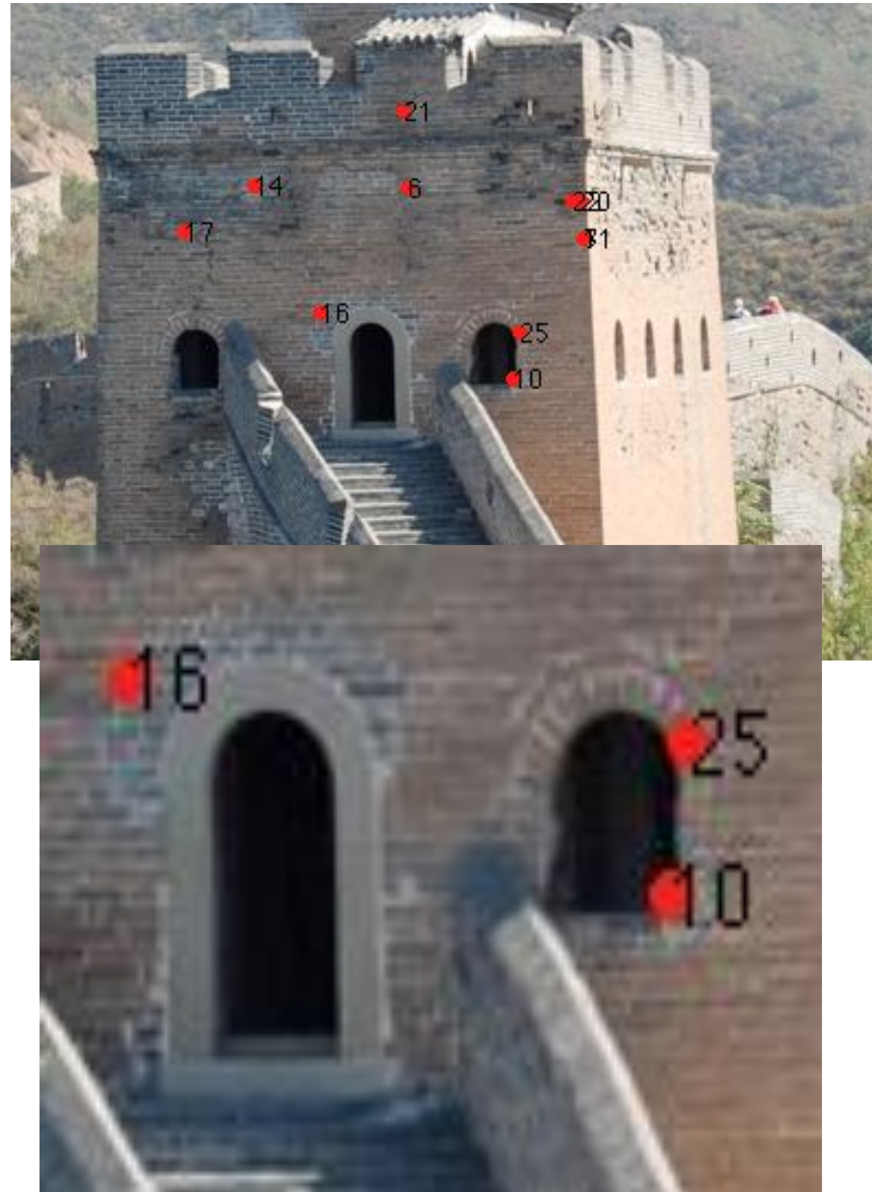
SIFT keypoints and descriptors

SIFT matching points in image 1

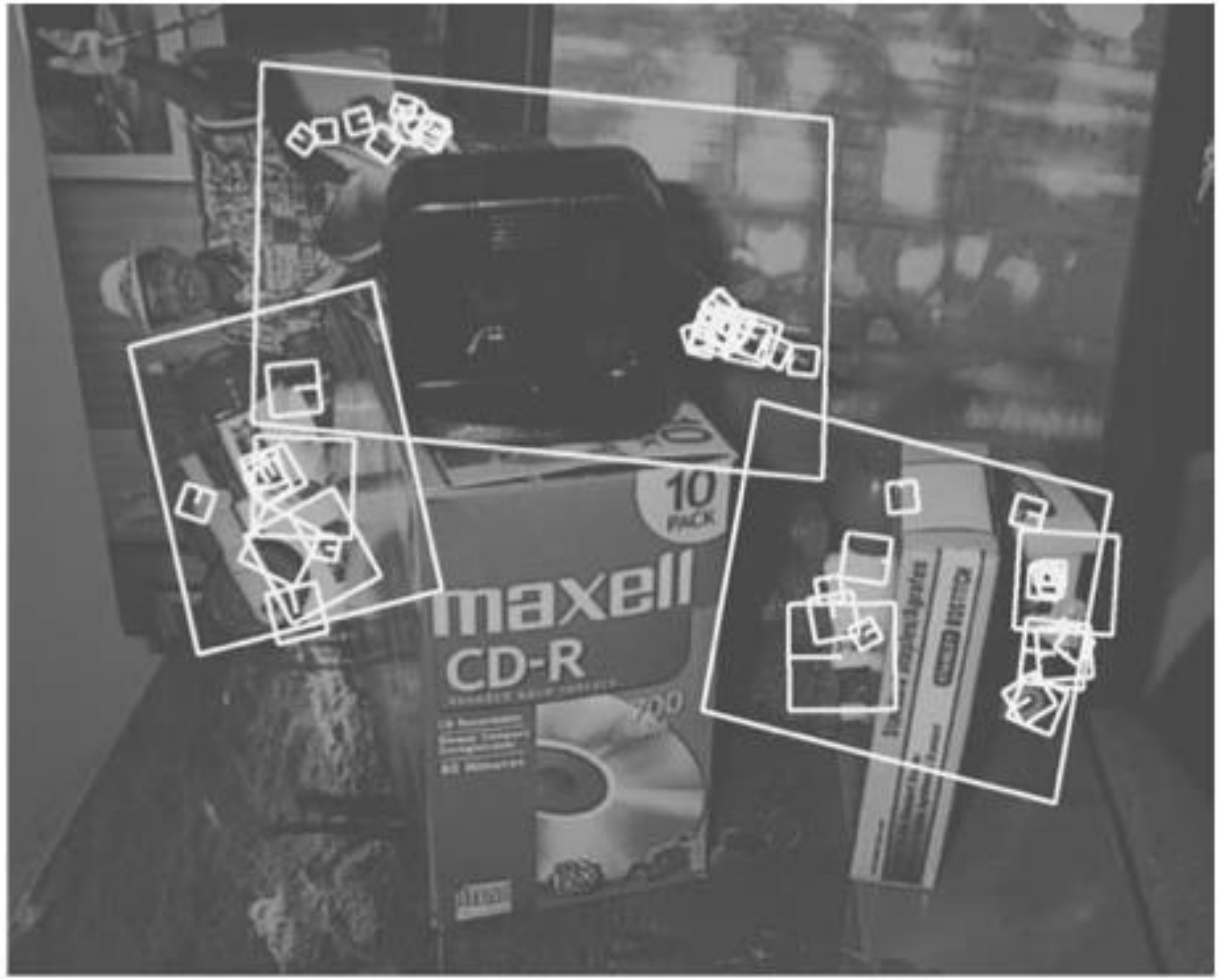
SIFT matching points in image 2

SIFT matching points in images 1 and 2





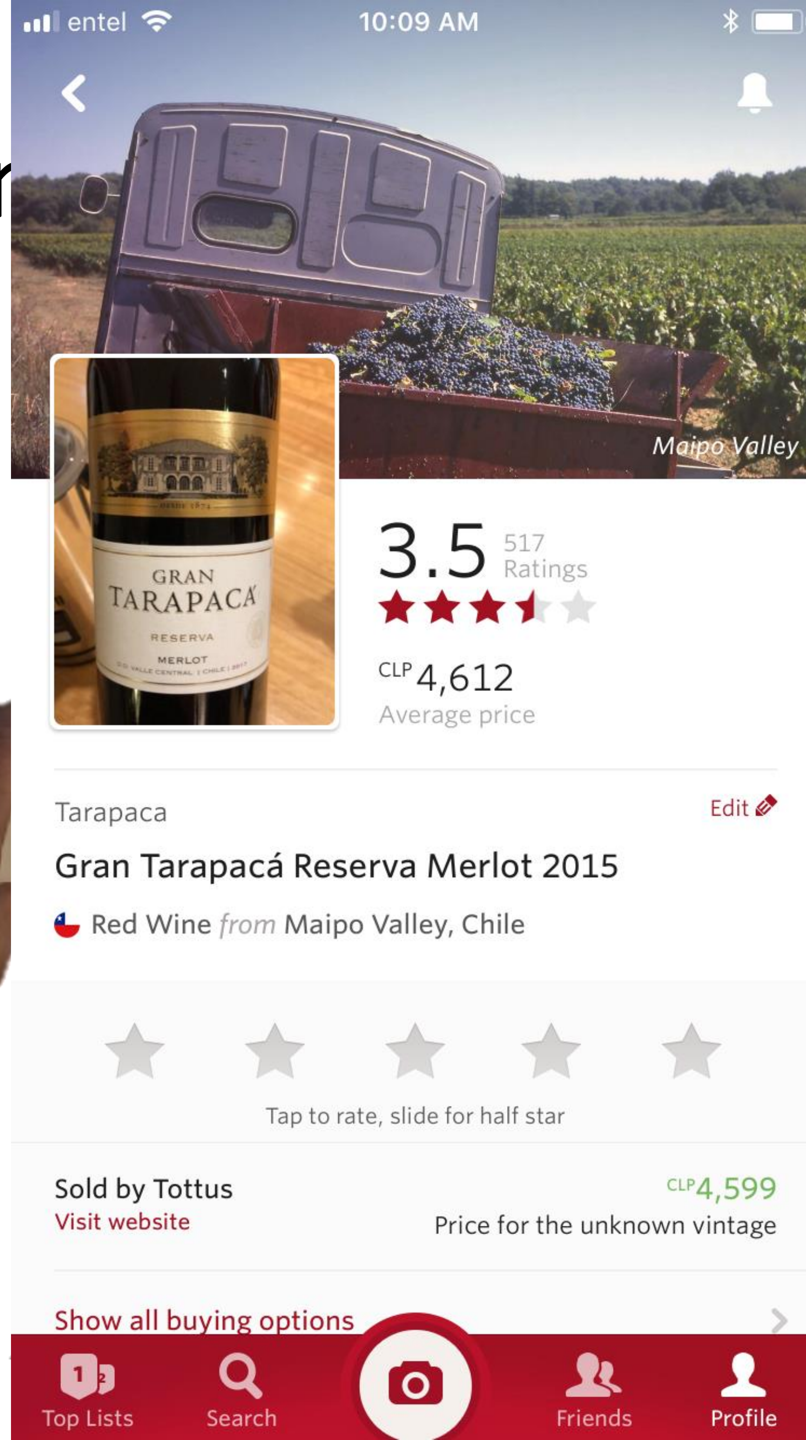
MATLAB Demo
PAT07_SIFT.m



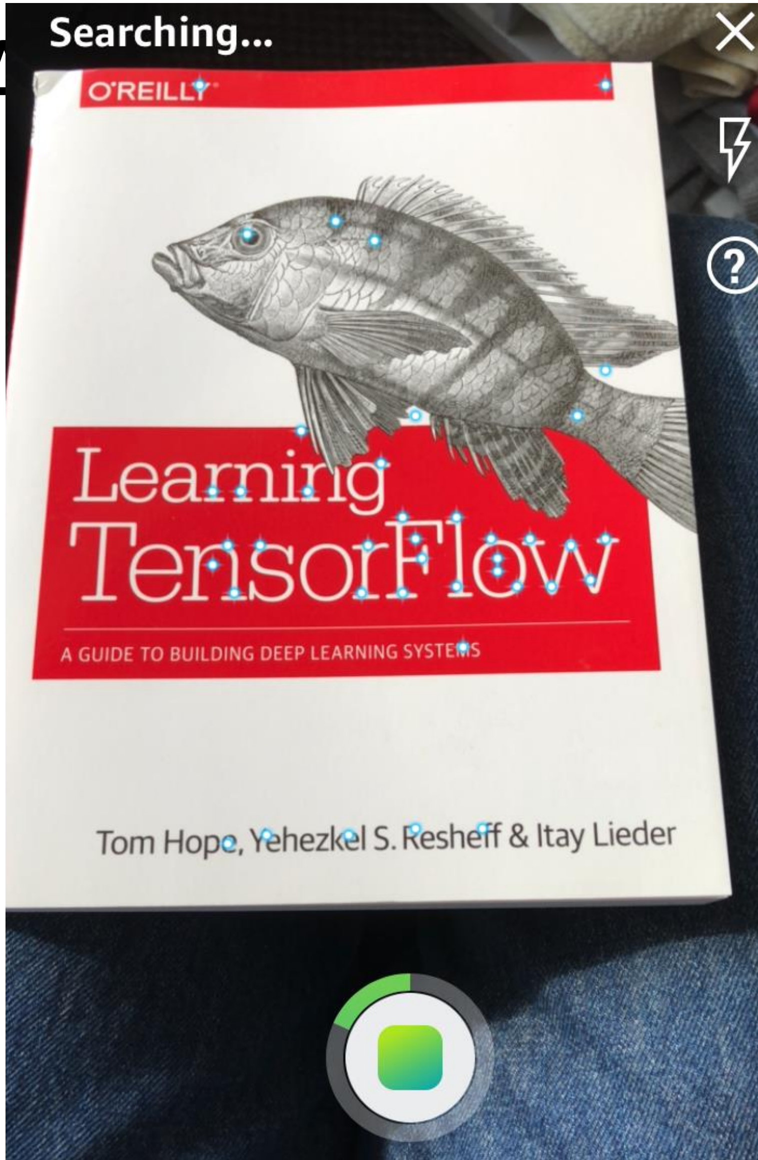
© Lowe

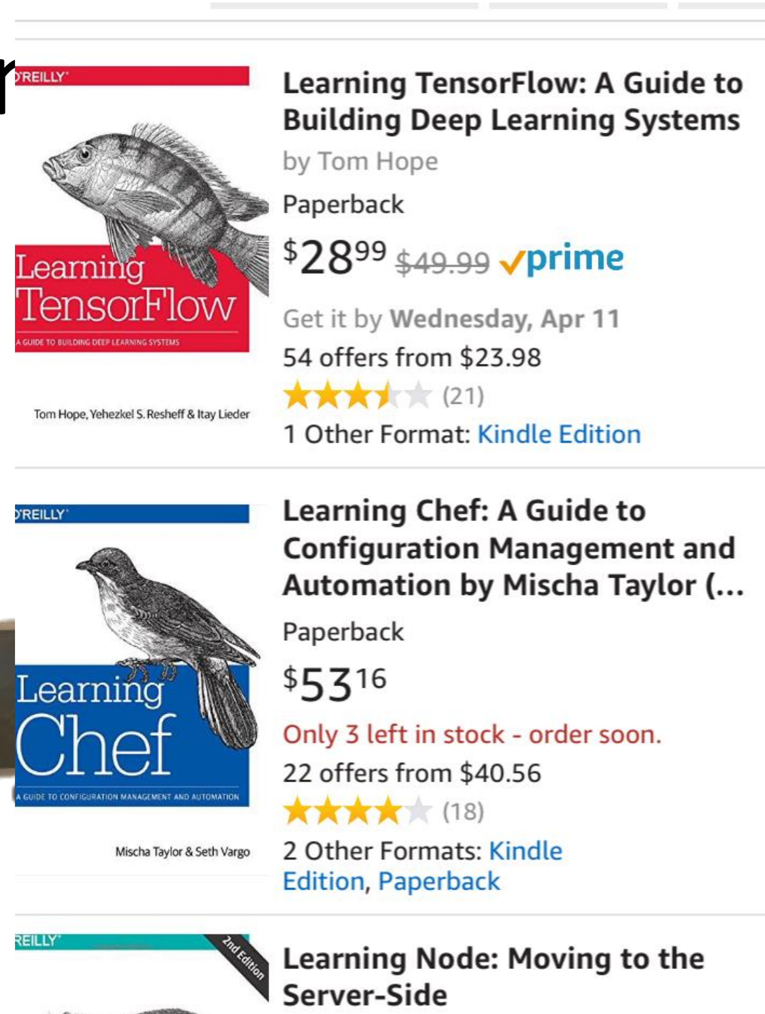
© Lowe

Applications: Vivir



A azor


Searching...
O'REILLY
Learning TensorFlow
A GUIDE TO BUILDING DEEP LEARNING SYSTEMS
Tom Hope, Yehezkel S. Resheff & Itay Lieder


Learning TensorFlow: A Guide to Building Deep Learning Systems
by Tom Hope
Paperback
$28.99 $49.99 prime
Get it by Wednesday, Apr 11
54 offers from $23.98
(21)
1 Other Format: Kindle Edition
Learning Chef: A Guide to Configuration Management and Automation by Mischa Taylor (...
Paperback
$53.16
Only 3 left in stock - order soon.
22 offers from $40.56
(18)
2 Other Formats: Kindle Edition, Paperback
Learning Node: Moving to the Server-Side